



# Modbus-Konfiguration und Registerbelegungen

2025 Oktober

# INDEX

<b><u>1. Einleitung</u></b>	<b>4</b>
1.1. Was ist Modbus?	5
1.2. Wozu wird Modbus verwendet?	5
1.3. Aufbau TCP-/RTU-Paket	6
<b><u>2. Registerbelegung</u></b>	<b>7</b>
2.1. Allgemeines	8
2.2. Statusregister	9
2.2.1. Lesen von Statusregistern	9
2.3. I/O Konfiguration / Anzahl der Kanäle	10
2.4. Digitale Ausgänge	11
2.4.1. Setzen und Zurücklesen digitaler Ausgänge (1 Kanal)	11
2.4.2. Setzen und Zurücklesen digitaler Ausgänge (16 Kanäle)	12
2.4.3. Bitweises Setzen digitaler Ausgänge (16 Kanäle)	13
2.4.4. Bitweises Zurücksetzen digitaler Ausgänge (16 Kanäle)	14
2.5. Digitale Eingänge	15
2.5.1. Lesen digitaler Eingänge (1 Kanal)	15
2.5.2. Lesen digitaler Eingänge (16 Kanäle)	16
2.5.3. Lesen von Eingangszustandsänderungen (16 Kanäle)	17
2.5.4. Eingangszähler Lesen (1 Kanal)	18
2.5.5. Eingangszähler Lesen und Zurücksetzen auf 0 (1 Kanal)	19
2.5.6. Intervall für Eingangsfiler Schreiben und lesen	20
2.6. TTL-Ein-/Ausgänge	22
2.6.1. Setzen und Zurücklesen der TTL-Richtung (8 Kanäle)	22
2.7. Analoge Ausgänge	23
2.7.1. D/A-Werte Schreiben und Zurücklesen	23
2.7.2. D/A-Modus Setzen und Zurücklesen	24
2.8. Analoge Eingänge	25
2.8.1. Lesen analoger Eingänge	25
2.8.2. A/D-Modus Schreiben und Zurücklesen	26

# INDEX

2.8.3. A/D-Filterlevel Schreiben und Zurücklesen	27
2.9. Timeout für digitale und analoge Ausgänge	28
2.9.1. Timeout-Modus Setzen	28
2.9.2. Timeout-Status Lesen	31
2.9.3. Timeout-Wert Setzen und Zurücklesen	32
2.9.4. Bitweises Setzen digitaler Ausgänge im Timeout-Fall (16 Kanäle)	33
2.9.5. Bitweises Zurücksetzen digitaler Ausgänge im Timeout-Fall (16 Kanäle)	34
<b><u>3. Anhang</u></b>	<b>35</b>
3.1. Revisionen	36
3.2. Urheberrechte und Marken	37

# Einleitung



# 1. Einleitung

## 1.1. Was ist Modbus?

Modbus ist eines der am weitesten verbreiteten Kommunikationsprotokolle in der Automatisierungstechnik. Es ist leicht verständlich und lässt sich auf Grund des offenen Protokolls herstellerübergreifend anwenden.

Der Datenaustausch basiert auf einer Master-Slave Struktur, wobei die Master Einheit die Anfragen initiiert und die Slave Einheiten auf dessen Befehle antworten.

Das Protokoll umfasst vier grundlegenden Datentypen:

- |                      |                                       |
|----------------------|---------------------------------------|
| 1. Coils             | -> Schreiben und Zurücklesen - 1 bit  |
| 2. Discrete Inputs   | -> Lesen - 1 bit                      |
| 3. Input Registers   | -> Lesen - 16 bit                     |
| 4. Holding Registers | -> Schreiben und Zurücklesen - 16 bit |

Die beiden wichtigsten Übertragungsarten sind Modbus RTU (seriell) und Modbus TCP (Ethernet basiert).

## 1.2. Wozu wird Modbus verwendet?

Die typischen Einsatzgebiete von Modbus finden sich in der Automatisierungstechnik, in Prozess- und Anlagensteuerungen, in der Gebäudeautomation und in Energie- und Umwelttechnik. Die Kommunikation zwischen einer SPS und angeschlossenen Sensoren, Aktoren oder Frequenzumrichtern sowie das Auslesen von Energiezählern oder anderen Messgeräten sind hier die häufigsten Anwendungsgebiete. Dies kann aus einem Mix an unterschiedlichen Herstellern mit Modbus-Geräten erfolgen.

### Vorteile von Modbus

- Offenes Protokoll
- Herstellerunabhängig einsetzbar
- Modbus TCP kann auf einer bestehenden Netzwerk-Infrastruktur angewendet werden

## 1.3. Aufbau TCP-/RTU-Paket

### Was macht ein Modbus-Paket?

Ein Modbus-Paket ist eine Nachricht vom Sender (Client/Master) an den Empfänger (Server/Slave) oder umgekehrt.

### Beispiel

Der Sender stellt eine Anfrage: z. B. „Lies mir 2 Holding Register ab Adresse 40001“.

Der Empfänger verarbeitet diese Anfrage und schickt die passende Antwort zurück: z. B. „Hier sind die beiden Werte: 1200 und 450“.

### Was enthält ein Modbus-Paket grundsätzlich?

- Adressinformationen → Welches Gerät ist gemeint (Slave-ID oder Unit Identifier).
- Funktionscode → Welche Aktion soll ausgeführt werden (z. B. Lesen, Schreiben).
- Daten → Welche Adresse(n) und wie viele Werte betroffen sind, bzw. die Werte selbst.
- Fehlerprüfung → bei TCP übernimmt das TCP/IP-Protokoll.

### Aufbau Modbus TCP - Paket

Ein TCP-Telegramm wird über Ethernet (Port 502) gesendet und besteht aus:

- MBAP-Header (7 Byte):
  - a. Transaction Identifier
  - b. Protocol Identifier (immer 0)
  - c. Length (Anzahl der folgenden Bytes)
  - d. Unit Identifier (ähnlich der Slave-Adresse, wichtig bei Gateways)
- Funktionscode – legt fest, was gemacht werden soll (z. B. Register lesen).
- Daten – Adressen, Werte oder Parameter.

# Registerbelegung



## 2. Registerbelegung

### 2.1. Allgemeines

Mit den Modbus-Registerbelegungen wird festgelegt, welche Daten an welcher Adresse in einem Gerät liegen.

Die DEDITEC Modbus-Register beginnen immer bei 0 (base = 0).

#### Allgemeine Registeraufteilung

Modbus Register (dez)	Daten-Typ	Zugriff	Data size (Bit)	Beschreibung
0 ... 9999	Coil	write / read	1	Digitale Ausgänge Schreiben und Zurücklesen
10000 ... 19999	Discrete Inputs	read	1	Digitale Eingänge lesen
30000 ... 39999	Input Register	read	16	Halteregister lesen
40000 ... 49999	Holding Register	write / read	16	Register Schreiben und Zurücklesen

Mit der richtigen Registerbelegung weiß man also genau, wo welche Information im Gerät gespeichert ist und kann diese gezielt auslesen, überwachen oder verändern.

## 2.2. Statusregister

### 2.2.1. Lesen von Statusregistern

Mithilfe der folgenden Adressen können Informationen zu dem Modul abgerufen werden.

Die zurückgelesenen Parameter sind in der delib\_defines.h Datei definiert.

Die Datei finden Sie nach der Installation unseres Software-Pakets an folgender Stelle:

32-Bit: C:\Program Files (x86)\DEDITEC\DELIB\include

64-Bit: C:\Program Files\DEDITEC\DELIB64\include

#### Input Register (3xxxx) - Base 0

Modbus Register (dez)	Register adresse (hex)	Zugriff	Data size (Bit)	Beschreibung
39200	0x23F0	read	16	Firmware-Version
39168	0x23D0	read	32	Software Feature
39170	0x23D2	read	32	Hardware Interface
39172	0x23D4	read	32	Module-Config Feature
39174	0x23D6	read	32	EC Events Feature
39176	0x23D8	read	32	TCP Feature
39178	0x23DA	read	32	HW-Group
39180	0x23DC	read	32	SW-Class

Wird bei der Firmware-Version der Wert 0101 gelesen, handelt es sich um die Versionsnummer 1.01.

## 2.3. I/O Konfiguration / Anzahl der Kanäle

Der folgenden Tabelle können Sie die Registeradresse entnehmen, die Sie zum Auslesen der Kanalanzahl benötigen.

### Input Register (3xxxx) - Base 0

Modbus Register (dez)	Register adresse (hex)	Zugriff	Data size (Bit)	Beschreibung
38960	0x2300	read	16	Digitale Ausgangskanäle
38961	0x2301	read	16	Digitale Eingangskanäle
38962	0x2302	read	16	Digitale Ein-/Ausgangskanäle (TTL)
38963	0x2303	read	16	Analoge Ausgangskanäle
38964	0x2304	read	16	Analoge Eingangskanäle
38965	0x2305	read	16	Stepperkanäle
38966	0x2306	read	16	Zähler für Eingangszustandsänderungen
38967	0x2307	read	16	Eingangszähler
38968	0x2308	read	16	Temperaturkanäle
38969	0x2309	read	16	48-Bit Eingangszähler
38970	0x230A	read	16	Pulsgenerator Ausgangskanäle
38971	0x230B	read	16	PWM Ausgangskanäle

## 2.4. Digitale Ausgänge

Die Register zum Ansteuern der Ausgangsmodule können nicht nur beschrieben, sondern auch zurückgelesen werden. Somit lässt sich der aktuelle Ausgangszustand eines (oder mehrerer Module) zu einem späteren Zeitpunkt abfragen. Beim versehentlichen Beenden der Software oder gar eines Absturzes ist der Zustand der Ausgangsmodule erfassbar. Ein laufender Prozess muss somit nicht zurückgesetzt werden.

Mit den folgenden Registeradressen, können die digitalen Ausgänge des Moduls gesteuert werden.

### 2.4.1. Setzen und Zurücklesen digitaler Ausgänge (1 Kanal)

Coils (0xxxx) - Base 0

Modbus Register (dez)	Register adresse (hex)	Zugriff	Data size (Bit)	Beschreibung
00000	0x0000	r/w	1	Setzen und Zurücklesen von Kanal 1
00001	0x0001	r/w	1	Setzen und Zurücklesen von Kanal 2
...	...			
00255 *	0x00FF *	r/w	1	Setzen und Zurücklesen von Kanal 256

\* der maximale Adressbereich ist abhängig von der Kanalanzahl des Moduls

Diese Register dienen zum Setzen und Zurücklesen von einzelnen digitalen Ausgängen.

Wird der Registeradresse der Wert 1 übergeben, wird der jeweilige Kanal eingeschaltet. Mit dem Wert 0 wird der Kanal ausgeschaltet.

Beim Zurücklesen der Registeradresse ist der Rückgabewert 0 oder 1.

#### Beispiel:

Auf Register 0x0001 wird der Wert 1 geschrieben. Kanal 2 wird eingeschaltet.

Auf Register 0x000F wird der Wert 0 geschrieben. Kanal 16 wird ausgeschaltet.

## 2.4.2. Setzen und Zurücklesen digitaler Ausgänge (16 Kanäle)

### Holding Register (4xxxx) - Base 0

Modbus Register (dez)	Register adresse (hex)	Zugriff	Data size (Bit)	Beschreibung
41024	0x0400	r/w	16	Setzen und Zurücklesen von Kanal 1 - 16
41025	0x0401	r/w	16	Setzen und Zurücklesen von Kanal 17 - 32
...	...			
41039 *	0x040F *	r/w	16	Setzen und Zurücklesen von Kanal 241 - 256

\* der maximale Adressbereich ist abhängig von der Kanalanzahl des Moduls

Mit diesem Register werden gleichzeitig 16 digitale Ausgänge gesetzt.

Beim Lesen des Registers wird der aktuelle Bitwert als Hexadezimalzahl zurückgelesen.

#### Beispiel:

Auf Adresse 0x0400 wird der Wert 0xff00 geschrieben.

Ausgänge 1-8 werden auf 0 gesetzt.

Ausgänge 9-16 werden auf 1 gesetzt.

### 2.4.3. Bitweises Setzen digitaler Ausgänge (16 Kanäle)

#### Holding Register (4xxxx) - Base 0

Modbus Register (dez)	Register adresse (hex)	Zugriff	Data size (Bit)	Beschreibung
41056	0x0420	write	16	Setzen von Kanal 1 - 16
41057	0x0401	write	16	Setzen von Kanal 17 - 32
...	...			
41071 *	0x042F *	write	16	Setzen von Kanal 241 - 256

\* der maximale Adressbereich ist abhängig von der Kanalanzahl des Moduls

Mit diesem Register können Ausgänge gezielt auf 1 geschaltet werden, ohne die Zustände der benachbarten Ausgänge zu ändern.

Nur die Bits mit einer Wertigkeit von 1 im Übergabewert werden vom Befehl berücksichtigt.

#### Beispiel:

Auf Adresse 0x0420 wird der Wert 0x01 geschrieben. Ausgang 0 wird auf 1 gesetzt, der Zustand von Ausgang 1-31 bleibt unberührt.

Auf Adresse 0x0420 wird der Wert 0xff geschrieben. Ausgang 0-7 wird auf 1 gesetzt, der Zustand von Ausgang 8-31 bleibt unberührt.

Auf Adresse 0x0420 wird der Wert 0xff000000 geschrieben. Ausgang 23-31 wird auf 1 gesetzt, der Zustand von Ausgang 0-22 bleibt unberührt.

## 2.4.4. Bitweises Zurücksetzen digitaler Ausgänge (16 Kanäle)

### Holding Register (4xxxx) - Base 0

Modbus Register (dez)	Register adresse (hex)	Zugriff	Data size (Bit)	Beschreibung
41088	0x0440	write	16	Zurücksetzen von Kanal 1 - 16
41089	0x0441	write	16	Zurücksetzen von Kanal 17 - 32
...	...			
41103 *	0x044F *	write	16	Zurücksetzen von Kanal 241 - 256

\* der maximale Adressbereich ist abhängig von der Kanalanzahl des Moduls

Mit diesem Register können Ausgänge gezielt auf 0 geschaltet werden, ohne die Zustände der benachbarten Ausgänge zu ändern.

#### Beispiel:

Auf Adresse 0x0440 wird der Wert 0x1 geschrieben. Ausgang 0 wird auf 0 gesetzt, der Zustand von Ausgang 1-31 bleibt unberührt

Auf Adresse 0x0440 wird der Wert 0xff geschrieben. Ausgang 0-7 wird auf 0 gesetzt, der Zustand von Ausgang 8-31 bleibt unberührt

Auf Adresse 0x0440 wird der Wert 0xff000000 geschrieben. Ausgang 23-31 wird auf 0 gesetzt, der Zustand von Ausgang 0-22 bleibt unberührt

## 2.5. Digitale Eingänge

Die logischen Zustände der Spannungspegel an den Eingangsmodulen können über die folgenden Register ausgelesen werden.

### 2.5.1. Lesen digitaler Eingänge (1 Kanal)

Discret Inputs (1xxxx) - Base 0

Modbus Register (dez)	Register adresse (hex)	Zugriff	Data size (Bit)	Beschreibung
10256	0x0100	read	1	Lesen von Kanal 1
10257	0x0101	read	1	Lesen von Kanal 2
...	...			
10511 *	0x01FF *	read	1	Lesen von Kanal 256

\* der maximale Adressbereich ist abhängig von der Kanalanzahl des Moduls

Diese Register dienen zum Lesen eines einzelnen digitalen Eingangs.

Rückgabewert ist der Zustand des Eingangs (0/1).

#### Beispiel:

Register 0x0101 gibt den Zustand von Kanal 2 wieder.

Register 0x010F gibt den Zustand von Kanal 16 wieder.

## 2.5.2. Lesen digitaler Eingänge (16 Kanäle)

### Input Register (3xxxx) - Base 0

Modbus Register (dez)	Register adresse (hex)	Zugriff	Data size (Bit)	Beschreibung
30000	0x0000	read	16	Lesen von Kanal 1 - 16
30001	0x0001	read	16	Lesen von Kanal 17 - 32
...	...			
30015 *	0x000F *	read	16	Lesen von Kanal 241 - 256

\* der maximale Adressbereich ist abhängig von der Kanalanzahl des Moduls

Diese Register dienen zum Lesen von 16 digitalen Eingängen.

Beim Lesen des Registers wird der aktuelle Bitwert als Hexadezimalzahl zurückgegeben.

#### Beispiel:

Register 0x0001 gibt die Zustände von Kanal 17 bis 32 wieder.

Register 0x0004 gibt die Zustände von Kanal 65 bis 80 wieder.

### 2.5.3. Lesen von Eingangszustandsänderungen (16 Kanäle)

#### Input Register (3xxxx) - Base 0

Modbus Register (dez)	Register adresse (hex)	Zugriff	Data size (Bit)	Beschreibung
30016	0x0010	read	16	Lesen der Eingangszustandsänderung von Kanal 1 - 16
30017	0x0011	read	16	Lesen der Eingangszustandsänderung von Kanal 17 - 32
...	...			
30031 *	0x001F *	read	16	Lesen der Eingangszustandsänderung von Kanal 241 - 256

\* der maximale Adressbereich ist abhängig von der Kanalanzahl des Moduls

Dieses Register liest die Eingangszustandsänderungen von 16 Kanälen aus.

#### **Achtung!**

Beim Lesen der Eingangszustandsänderung werden diese direkt zurückgesetzt (auf 0)

#### **Beispiel:**

Register 0x0010 gibt die Eingangszustandsänderung-Merker von Kanal 17 bis 32 wieder und setzt diese zurück.

Register 0x0014 gibt die Eingangszustandsänderung-Merker von Kanal 65 bis 80 wieder und setzt diese zurück.

## 2.5.4. Eingangszähler Lesen (1 Kanal)

### Input Register (3xxxx) - Base 0

Modbus Register (dez)	Register adresse (hex)	Zugriff	Data size (Bit)	Beschreibung
30512	0x0200	read	16	Lesen von Eingangszähler Kanal 1
30513	0x0201	read	16	Lesen von Eingangszähler Kanal 2
...	...			
30527 *	0x02FF *	read	16	Lesen von Eingangszähler Kanal 256

\* der maximale Adressbereich ist abhängig von der Kanalanzahl des Moduls

Diese Register dienen zum Lesen der Eingangszähler (16-Bit Zähler).  
Beim Lesen des Registers wird der aktuelle Zählerwert zurückgegeben.

#### Beispiel:

Register 0x0201 gibt den Eingangszähler von Kanal 2 wieder.

Register 0x0204 gibt den Eingangszähler von Kanal 5 wieder.

## 2.5.5. Eingangszähler Lesen und Zurücksetzen auf 0 (1 Kanal)

### Input Register (3xxxx) - Base 0

Modbus Register (dez)	Register adresse (hex)	Zugriff	Data size (Bit)	Beschreibung
30768	0x0300	read	16	Lesen und Zurücksetzen von Kanal 1
30769	0x0301	read	16	Lesen und Zurücksetzen von Kanal 2
...	...			
30783 *	0x03FF *	read	16	Lesen und Zurücksetzen von Kanal 256

\* der maximale Adressbereich ist abhängig von der Kanalanzahl des Moduls

Diese Register dienen zum Lesen und Zurücksetzen der Eingangszählern.  
Beim Lesen des Registers wird der aktuelle Zählerwert zurückgegeben und dann auf 0 gesetzt.

#### Beispiel:

Register 0x0301 liest den Eingangszähler von Kanal 2 wieder und setzt diesen zurück auf 0.

Register 0x0304 gibt den Eingangszähler von Kanal 5 wieder und setzt diesen zurück auf 0.

## 2.5.6. Intervall für Eingangsfiler Schreiben und lesen

### Intervall des Eingangszustandsänderungsfilters

Holding Register (4xxxx) - Base 0

Modbus Register (dez)	Register adresse (hex)	Zugriff	Data size (Bit)	Beschreibung
48704	0x2200	r/w	8	Setzen und Lesen des Zeitintervalls des FF-Filters

Register zum Setzen und Zurücklesen des Eingangszustandsänderungsfilters [ms], das das Zeitintervall für die Abtastung der Eingangszustandsänderungs-Flipflops angibt.

Unterstützt werden Impulszeiten zwischen 5 ms und 255 ms - ohne Angabe gilt der Standardwert von 100 ms.

#### Wichtig!

Dieses Register wird nur unterstützt, wenn das Modul digitale Eingänge besitzt.

## Eingangsfiter Intervall

### Holding Register (4xxxx) - Base 0

Modbus Register (dez)	Register adresse (hex)	Zugriff	Data size (Bit)	Beschreibung
48705	0x2201	r/w	8	Setzen und Lesen des Zeitintervalls des Eingangs-Filters

Register zum Setzen und Lesen eines Eingangsfilters in [ms], der das Zeitintervall angibt, in dem Störimpulse bei digitalen Eingangskanälen gefiltert werden.

Dieser Befehl unterstützt Impulszeiten zwischen 1 ms und 255 ms. Wird keine Zeit gesetzt, ist der Default-Wert 0 ms (=off).

### Wichtig!

Dieses Register wird nur unterstützt, wenn das Modul digitale Eingänge besitzt.

## 2.6. TTL-Ein-/Ausgänge

### 2.6.1. Setzen und Zurücklesen der TTL-Richtung (8 Kanäle)

Holding Register (4xxxx) - Base 0

Modbus Register (dez)	Register adresse (hex)	Zugriff	Data size (Bit)	Beschreibung
48752	0x2230	r/w	8	Setzt die Richtung der TTL-Ein-/Ausgänge

Dieses Register dient zum Setzen der Richtung von bis zu 64 hintereinanderliegenden TTL Ein/Ausgängen (8-Bit weise).

1-Bit repräsentiert dabei 8 TTL-Ein/Ausgänge. Wert 1 = output / 0 = input.

Beim Lesen des Registers wird der aktuelle 8-Bitwert als Hexadezimalzahl zurückgelesen.

Bit 0: Richtung von TTL 0-7	/ 1=Output, 0=Input
Bit 1: Richtung von TTL 8-15	/ 1=Output, 0=Input
Bit 2: Richtung von TTL 16-23	/ 1=Output, 0=Input
Bit 3: Richtung von TTL 24-31	/ 1=Output, 0=Input
Bit 4: Richtung von TTL 32-39	/ 1=Output, 0=Input
Bit 5: Richtung von TTL 40-47	/ 1=Output, 0=Input
Bit 6: Richtung von TTL 48-55	/ 1=Output, 0=Input
Bit 7: Richtung von TTL 56-63	/ 1=Output, 0=Input

#### Beispiel:

Auf Register 0x2230 wird der Wert 0x1 geschrieben. Setzt Kanal 0-7 auf Ausgang, die anderen auf Eingang.

Auf Register 0x2230 wird der Wert 0xC geschrieben. Setzt Kanal 16-31 auf Ausgang, die anderen auf Eingang.

Auf Register 0x2230 wird der Wert 0xFF geschrieben. Setzt Kanal 0-63 auf Ausgang.

## 2.7. Analoge Ausgänge

Mit den folgenden Registern lassen sich D/A-Werte und D/A-Modi setzen und zurücklesen.

### 2.7.1. D/A-Werte Schreiben und Zurücklesen

Holding Register (4xxxx) - Base 0

Modbus Register (dez)	Register adresse (hex)	Zugriff	Data size (Bit)	Beschreibung
42816	0x0B00	r/w	16	Setzen und Zurücklesen von Kanal 1
42817	0x0B01	r/w	16	Setzen und Zurücklesen von Kanal 2
...	...			
43071 *	0x0BFF *	r/w	16	Setzen und Zurücklesen von Kanal 256

\* der maximale Adressbereich ist abhängig von der Kanalanzahl des Moduls

Mit diesen Registern wird ein Datenwert an ein Kanal eines 16-Bit D/A-Wandlers übergeben.

Datenwert der geschrieben werden kann (16-Bit Datenwert -> Datenwertebereich: 0-65535).

Beim Zurücklesen des Registers, wird der aktuelle D/A-Wert des Kanals ausgelesen.

#### Beispiel:

Auf Register 0x0B01 wird der Wert 0xFFFF geschrieben.

Setzt den 1. Ausgang des D/A-Wandlers auf maximalen Wert des gewählten Modus.

Bei ausgewähltem Modus ADDA\_MODE\_UNIPOL\_10V wird der 1. Ausgang des D/A-Wandlers auf 10V gesetzt.

## 2.7.2. D/A-Modus Setzen und Zurücklesen

### Holding Register (4xxxx) - Base 0

Modbus Register (dez)	Register adresse (hex)	Zugriff	Data size (Bit)	Beschreibung
42560	0x0A00	r/w	8	Setzen und Zurücklesen des D/A-Modus von Kanal 1
42561	0x0A01	r/w	8	Setzen und Zurücklesen des D/A-Modus von Kanal 2
42815*	0x0aFF*	r/w	8	Setzen und Zurücklesen des D/A-Modus von Kanal 256

\* der maximale Adressbereich ist abhängig von der Kanalanzahl des Moduls

Mit diesem Register wird der Modus für einen D/A-Wandler gesetzt.

Die möglichen Spannungs- und Strommodi sind Modul abhängig und können dem Handbuch des Moduls unter "Technische Daten" entnommen werden.

Beim Zurücklesen des Registers, wird der aktuell gesetzte D/A-Modus des Kanals ausgelesen.

#### Beispiel:

Wird dem Register 0x0A41 der Wert 0x01 übergeben, wird der D/A-Modus von Kanal 2 auf "±5 V" gesetzt.

## 2.8. Analoge Eingänge

Mit den folgenden Registern lassen sich die A/D-Messwerte des Moduls auslesen und A/D-Modi und Filterlevel einstellen.

### 2.8.1. Lesen analoger Eingänge

Input Register (3xxxx) - Base 0

Modbus Register (dez)	Register adresse (hex)	Zugriff	Data size (Bit)	Beschreibung
32304	0x0900	read	16	Lesen von Kanal 1
32305	0x0901	read	16	Lesen von Kanal 2
...	...			
32559 *	0x09FF *	read	16	Lesen von Kanal 256

\* der maximale Adressbereich ist abhängig von der Kanalanzahl des Moduls

Dieses Register dient zum Lesen eines Datenwertes von einem Kanal des A/D-Wandlers.

#### Beispiel:

Register 0x0901 gibt den Messwert von Kanal 2 wieder.

Register 0x0904 gibt den Messwert von Kanal 5 wieder.

## 2.8.2. A/D-Modus Schreiben und Zurücklesen

### Holding Register (4xxxx) - Base 0

Modbus Register (dez)	Register adresse (hex)	Zugriff	Data size (Bit)	Beschreibung
42048	0x0800	r/w	8	Setzt den A/D-Modus für Kanal 1
42049	0x0801	r/w	8	Setzt den A/D-Modus für Kanal 2
...	...			
42303*	0x08FF*	r/w	8	Setzt den A/D-Modus für Kanal 256

\* der maximale Adressbereich ist abhängig von der Kanalanzahl des Moduls

Mit diesem Register wird der Modus für einen Kanal des A/D-Wandler gesetzt.

Die möglichen Spannungs- und Strommodi sind Modul abhängig und können dem Handbuch des Moduls unter "Technische Daten" entnommen werden.

Beim Zurücklesen des Registers, wird der aktuell gesetzte A/D-Modus des Kanals ausgelesen.

#### Beispiel:

Wird dem Register 0x0801 der Wert 0x41 übergeben, wird der A/D-Modus von Kanal 2 auf "±5 V" gesetzt.

Wird dem Register 0x0804 der Wert 0x80 übergeben, wird der A/D-Modus von Kanal 5 auf "0-20 mA" gesetzt.

### 2.8.3. A/D-Filterlevel Schreiben und Zurücklesen

#### Holding Register (4xxxx) - Base 0

Modbus Register (dez)	Register adresse (hex)	Zugriff	Data size (Bit)	Beschreibung
38736	0x2220	r/w	8	Setzen und Lesen des A/D-Filterlevels

Mit diesem Register lässt sich das A/D-Filterlevel Ihres A/D-Moduls für alle Kanäle festlegen.

Bei Modulen mit mehreren A/D-Submodulen, kann das Filterlevel für jedes Submodul separat eingestellt werden.

Das Filterprinzip basiert auf einer Mittelwertbildung aus einer Summe von Messungen. Je höher das Filterlevel gewählt wird, desto mehr Messungen werden zur Mittelwertbildung herangezogen. Bedeutet aber auch, dass sich die Abtastrate des A/D-Wandlers mit steigendem Filterlevel reduziert.

Je nach Modul, kann die Höhe des einstellbaren Filterlevels variieren.

Der Wert 0 deaktiviert den A/D-Filter.

Beim Zurücklesen des Registers, wird der aktuell gesetzte A/D-Filterlevel ausgelesen.

#### Beispiel:

Wird dem Register 2220 der Wert 5 übergeben, wird der A/D-Filterlevel auf 5 gesetzt.

Wird dem Register 2220 der Wert 0 übergeben, wird der A/D-Filterlevel deaktiviert.

## 2.9. Timeout für digitale und analoge Ausgänge

Mit den folgenden Registern kann die Timeout-Funktion bei digitalen und analogen Ausgängen eingestellt werden.

### 2.9.1. Timeout-Modus Setzen

Holding Register (4xxxx) - Base 0

Modbus Register (dez)	Register adresse (hex)	Zugriff	Data size (Bit)	Beschreibung
48720	0x2210	write	8	Setzt den Timeout-Modus

Mit diesem Register wird der Timeout-Modus gesetzt.

**Folgende Modi können gesetzt werden:**

Wert = 0 (Timeout deaktivieren)

Wert = 1 ("normalen" Timeout aktivieren)

Wert = 3 ("auto reactivate" Timeout aktivieren)

Wert = 5 ("secure outputs" Timeout aktivieren)

#### "normalen" Timeout

Dies ist der Timeout, den unsere Module schon seit 2009 besitzen. Vorgehensweise für den Timeout-Befehl: Der Timeout wird per Befehl aktiviert. Findet dann ein sogenanntes Timeout-Ereignis statt(Pause zwischen zwei Zugriffen auf das Modul ist grösser, als die erlaubte Timeout-Zeit) passiert folgendes:

- Alle Ausgänge werden ausgeschaltet.
- Der Timeout-Status geht auf "2".

- Die Timeout-LED geht an (bei Modulen, die solch einen Status haben) Weitere Zugriffe auf die Ausgänge sind dann weiterhin möglich, aber der Timeout ist nicht weiter aktiv. Erst wieder, wenn er wieder aktiviert wurde.

### **"auto reactivate" Timeout**

Dies ist ein seit 2021 implementierter Timeout-Modus, der nach Auftreten des Timeout-Ereignisses den Timeout automatisch wieder aktiviert. Vorgehensweise für den Timeout-Befehl: Der Timeout wird per Befehl aktiviert. Findet dann ein sogenanntes Timeout-Ereignis statt (Pause zwischen zwei Zugriffen auf das Modul ist grösser, als die erlaubte Timeout-Zeit) passiert folgendes:

- Alle Ausgänge werden ausgeschaltet.
- Der Timeout-Status geht auf "4".
- Die Timeout-LED geht an (bei Modulen, die solch einen Status haben) Weitere Zugriffe auf die Ausgänge sind dann weiterhin möglich. UND der Timeout ist weiter aktiv. Bei erneuter Zeitüberschreitung der Timeout-Zeit werden die Ausgänge wieder ausgeschaltet.

### **"secure outputs" Timeout**

Dies ist ein seit 2021 implementierter Timeout-Modus, der nach Auftreten des Timeout-Ereignisses einen Schreibenden Zugriff auf die Ausgänge verhindert. Somit wird sichergestellt, dass die Software erst einmal einen "sicheren" Zustand der Ausgänge wiederherstellen muss, da der Timeout Mechanismus des Moduls die Ausgänge auf vordefinierte Werte verändert hat. Vorgehensweise für den Timeout-Befehl: Der Timeout wird per Befehl aktiviert. Findet dann ein sogenanntes Timeout-Ereignis statt (Pause zwischen zwei Zugriffen auf das Modul ist grösser, als die erlaubte Timeout-Zeit) passiert folgendes:

- Alle Ausgänge werden ausgeschaltet.
- Der Timeout-Status geht auf "6".
- Die Timeout-LED geht an (bei Modulen, die solch einen Status haben) Weitere Zugriffe auf die Ausgänge sind NICHT möglich. Erst nach erneutem Aktivieren des Timeouts oder Deaktivieren des Timeouts können die Ausgänge beschrieben werden.

## 2.9.2. Timeout-Status Lesen

### Holding Register (4xxxx) - Base 0

Modbus Register (dez)	Register adresse (hex)	Zugriff	Data size (Bit)	Beschreibung
48721	0x2211	read	8	Lesen des Timeout-Status

Mit diesem Register wird der Timeout-Status gelesen.

Ein Timeout-Ereignis hat dann stattgefunden, wenn der Timeout aktiviert wurde und für mindestens die Zeit des Timeout-Wertes kein Datenverkehr stattfand.

Return = 0 (Timeout ist deaktiviert)

#### Werte für "normalen" Timeout Timeout Status lesen

Return = 1 (Timeout "normal" ist aktiviert)

Return = 2 (Timeout "normal" hat stattgefunden)

#### Werte für "auto reactivate" Timeout

Return = 3 (Timeout "auto reactivate" ist aktiviert)

Return = 4 (Timeout "auto reactivate" hat ein oder mehrmals stattgefunden)

#### Werte für "secure" Timeout

Return = 5 (Timeout "secure outputs" ist aktiviert)

Return = 6 (Timeout "secure outputs" hat stattgefunden. In diesem Status wird ein schreiben auf die Outputs verhindert)

### 2.9.3. Timeout-Wert Setzen und Zurücklesen

#### Holding Register (4xxxx) - Base 0

Modbus Register (dez)	Register adresse (hex)	Zugriff	Data size (Bit)	Beschreibung
48722	0x2212	r/w	16	Setzt den Timeout-Wert

Der Timeout-Wert ist das Produkt aus 16 Bit-Wert \* 100 Millisekunden (die Einheit ist also 100 ms).

Der zulässige Wertebereich der Zeitangabe liegt zwischen 0,1 Sekunden und 6553 Sekunden.

Beim Zurücklesen des Registers, wird der aktuell gesetzte Timeout-Wert ausgelesen.

## 2.9.4. Bitweises Setzen digitaler Ausgänge im Timeout-Fall (16 Kanäle)

### Holding Register (4xxxx) - Base 0

Modbus Register (dez)	Register adresse (hex)	Zugriff	Data size (Bit)	Beschreibung
41248	0x04E0	r/w	16	Setzen und Lesen der Ausgänge 1 - 16 im Timeout-Fall
41249	0x04E1	r/w	16	Setzen und Lesen der Ausgänge 17 - 32 im Timeout-Fall
...	...			
41039 *	0x04EF *	r/w	16	Setzen und Lesen der Ausgänge 241 - 256 im Timeout-Fall

\* der maximale Adressbereich ist abhängig von der Kanalanzahl des Moduls

Dieses Register bestimmt die Ausgänge, die bei einem Timeout gesetzt werden sollen.

Beim Lesen des Registers wird der aktuelle Bitwert als Hexadezimalzahl zurückgelesen.

#### Beispiel:

Auf Adresse 0x04E0 wird der Wert 0x02 geschrieben. Kanal 2 wird gesetzt, alle anderen Kanäle bleiben unverändert.

Auf Adresse 0x04E0 wird der Wert 0x03 geschrieben. Kanal 1 und 2 werden gesetzt, alle anderen Kanäle bleiben unverändert.

Auf Adresse 0x04E0 wird der Wert 0x04 geschrieben. Kanal 3 wird gesetzt, alle anderen Kanäle bleiben unverändert.

## 2.9.5. Bitweises Zurücksetzen digitaler Ausgänge im Timeout-Fall (16 Kanäle)

Holding Register (4xxxx) - Base 0

Modbus Register (dez)	Register adresse (hex)	Zugriff	Data size (Bit)	Beschreibung
41216	0x04C0	r/w	16	Zurücksetzen der Ausgänge 1 - 16 im Timeout-Fall
41217	0x04C1	r/w	16	Zurücksetzen der Ausgänge 17 - 32 im Timeout-Fall
...	...			
41231 *	0x04CF *	r/w	16	Zurücksetzen der Ausgänge 241 - 256 im Timeout-Fall

\* der maximale Adressbereich ist abhängig von der Kanalanzahl des Moduls

Dieses Register bestimmt die Ausgänge, die bei einem Timeout ausgeschaltet werden sollen.

Beim Lesen des Registers wird der aktuelle Bitwert als Hexadezimalzahl zurückgelesen.

### Beispiel:

Auf Adresse 0x04C0 wird der Wert 0x02 geschrieben. Im Timeout-Fall wird Ausgang 2 zurückgesetzt, alle anderen Kanäle bleiben unverändert.

Auf Adresse 0x04C0 wird der Wert 0x03 geschrieben. Im Timeout-Fall werden Ausgang 1 und 2 zurückgesetzt, alle anderen Kanäle bleiben unverändert.

Auf Adresse 0x04C0 wird der Wert 0x04 geschrieben. Im Timeout-Fall wird Ausgang 3 zurückgesetzt, alle anderen Kanäle bleiben unverändert.

# Anhang



## **3. Anhang**

### **3.1. Revisionen**

1.0                      Erste DEDITEC Anleitung

### 3.2. Urheberrechte und Marken

Modbus ist eine registrierte Marke von Schneider Electric.

Linux ist eine registrierte Marke von Linus Torvalds.

USB ist eine registrierte Marke von USB Implementers Forum Inc.

LabVIEW ist eine registrierte Marke von National Instruments.

Intel ist eine registrierte Marke von Intel Corporation.

AMD ist eine registrierte Marke von Advanced Micro Devices, Inc.

ProfiLab ist eine registrierte Marke von ABACOM Ingenieurbüro GbR.

ispVM System ist eine registrierte Marke von Lattice Semiconductor Corporation

Windows, Visual-C/C++, -C#, -Basic, -Basic.NET und Visual-Studio sind registrierte Marken von Microsoft Corporation.

Delphi ist eine registrierte Marke von Borland Software Corporation.

Java ist eine registrierte Marke von Oracle Corporation.